

# **TDK-dolgozat**

Dénes Botond

# Vízközeli hulladéklerakók megbízható detektálása multispektrális műholdfelvételek segítségével

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI TANSZÉK



*Szerző:*

Dénes Botond  
programtervező informatikus MSc  
2. évfolyam

*Témavezető:*

Cserép Máté  
egyetemi tanársegéd

Budapest, 2024

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>3</b>
1.0.1. A kutatólabor eddigi eredményei . . . . .	4
<b>2. Kapcsolódó kutatások</b>	<b>5</b>
2.0.1. Hulladékdetectálási módszerek . . . . .	5
2.0.2. Random Forest . . . . .	7
<b>3. Elemzés és tervezés</b>	<b>9</b>
3.0.1. Kutatási cél . . . . .	9
3.0.2. Műhold specifikációk . . . . .	9
3.0.3. Használt indexek . . . . .	9
<b>4. Betanítás</b>	<b>11</b>
4.0.1. Tanítóadatok . . . . .	11
4.0.2. Tanítási paraméterek . . . . .	12
4.0.3. Főkomponens analízis (PCA) . . . . .	13
4.0.4. Nyári és téli adatokra való lebontás . . . . .	14
<b>5. Tesztelés</b>	<b>16</b>
5.0.1. A teszthalmaz . . . . .	16
5.0.2. Teljesítmény mérési módszerek . . . . .	18
<b>6. Megvalósítás és alkalmazás</b>	<b>19</b>
6.0.1. A meglevő alkalmazások bővítése . . . . .	19
6.0.2. A Tiszta-Tisza alkalmazás . . . . .	19
6.0.3. Közös metszet . . . . .	23
<b>7. Összefoglalás és eredmények</b>	<b>25</b>
<b>Köszönetnyilvánítás</b>	<b>26</b>

<b>A. Szimulációs eredmények</b>	<b>27</b>
<b>Irodalomjegyzék</b>	<b>28</b>
<b>Ábrajegyzék</b>	<b>31</b>
<b>Táblázatjegyzék</b>	<b>32</b>
<b>Algoritmusjegyzék</b>	<b>33</b>
<b>Forráskódjegyzék</b>	<b>34</b>

# 1. fejezet

## Bevezetés

szinkronizálni az  
absztrakttal

A hulladékszennyezés komoly problémát jelent a természet számára [1]. Emiatt számos szervezet mozdul abba az irányba, hogy tisztábbá tegye a bolygónkat. Egy ilyen szervezet a PET Kupa, akik folyómenti hulladékgyűjtéssel foglalkoznak elsősorban Magyarországon, de figyelmük kiterjed a szomszédos országokra is. Az egyik nagy kihívás a szemétgyűjtésben a hulladékkal szennyezett területeknek a hatékony megtalálása. Sok erőforrást igényel a hulladéklerakók megtalálása a folyók mentén, hiszen sokszor járművel kell valakinek végig haladnia egy hosszabb területen, azért, hogy felmérje, hogy hol van hulladék. A folyók árterén elhelyezett hulladékok még nagyobb problémát jelentenek, hiszen dagály idejében a hulladékot elmosa a víz és ez a folyó további szakaszaira lesz szétszórva miközben nagy károkat okoz a folyó élővilágának, illetve szennyezi a folyóvizet [2, 3]. Emiatt szükségünk van olyan eszközökre, melyekkel hamar lehet detektálni a szennyezett területeket, hogy ezeket minél hatékonyabban meglehessen tisztítani. Ennek fényében a PET Kupa felkereste az egyetemünket azzal a kéréssel, hogy olyan eszközöket fejlesszünk le, melyek automatikusan képesek lesznek hulladékot detektálni a folyók mentén.

A dolgozatomban bemutatok egy Random Forest modellt [4], mely a kutatólaborban már lefejlesztett modellre épül [5]. A bemutatott modell javít a korábbi megoldás pontosságán, illetve nagyobb megbízhatósággal találja meg a hulladékot a folyókon és a folyók mentén. A modell eredményei integrálásra kerülnek a Tiszta Tisza webalkalmazásba, ahol több napon keresztül történő detektálás eredménye lesz összesítve és megjelenítve a felhasználók számára. Ezen felül a dolgozatban tárgyalni fogok más kutatást is, mely a hulladékdetektálás problémájával foglalkozik. Ezen kívül kiegészíttem a kutatólabor meglevő szoftveres eszközeit annak érdekében, hogy a laborban zajló

Hivatkozni a  
weboldalra

munka gördülékenyebb legyen.

A kutatás hozzáadott terméke egy olyan adathalmaz, mely alkalmas más hulladékdetektálási modellek betanítására is. Az adathalmaz elsősorban szárazföldi romániai hulladéklerakókról készített PlanetScope műholdfelvételeket tartalmaz, melyek kézzel voltak annotálva. Az adatok georeferálva vannak, így ezeket könnyen meg lehet vizsgálni, illetve ki lehet egészíteni.

Továbbá bemutatom, hogy milyen módszerekkel próbáltam tovább javítani a modell eredményein. Ilyen módszer például a főkomponens analízis, a képnormálizálás, illetve az évszakokra bontás.

### 1.0.1. A kutatólabor eddigi eredményei

A térinformatikai kutatólaborban már fejlesztésre került egy szerveralkalmazás, mely minden nap a Planet-ről letölти a legfrissebb felvételeket a vizsgált területekről, és lefuttatja ezeken a képeken az akkori modellt. Ezen felül készült egy webalkalmazás is, ami erről a szerverről letölти az eredményeket, és megjeleníti ezeket, összehasonlításra. A kutatólabor rendelkezik egy asztali alkalmazással is, mellyel hatékonyan elő lehet állítani tanítóadatokat. A kutatásom elősegítéséhez ezeket az alkalmazásokat használtam, illetve bővítettem a 6.0.1 fejezetben leírtak szerint.

## **2. fejezet**

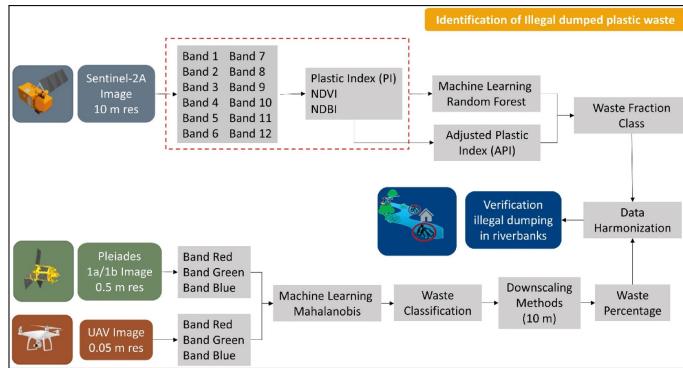
# **Kapcsolódó kutatások**

### **2.0.1. Hulladékdetektálási módszerek**

Számos kutatás készült már a hulladékdetektálás témajában. Az én kutatásom a Térinformatikai kutatólabor munkájára épül [5], ahol egy Random Forest modell került betanításra, hulladékdetektálás céljából. A kutatásban PlanetScope és Sentinel-2 műholdfelvételeket használtak. Ez a cikk rakta le az alapjait az én kutatásomnak is, melyben ezeken az eredményeken javítok. A cikkben további lehetséges munkaként említésre kerül a modell több adattal való tanítása, illetve a képfeldolgozás gyorsítása. A dolgozatom minden feladattal foglalkozik, A dolgozatomban csak a PlanetScope felvételek kerülnek felhasználásra, mivel a magasabb felbontású felvételek könnyebben lehetővé teszik a tanítóadatok előállítását, hiszen jobban lehet látni a hulladéklerakókat rajtuk.

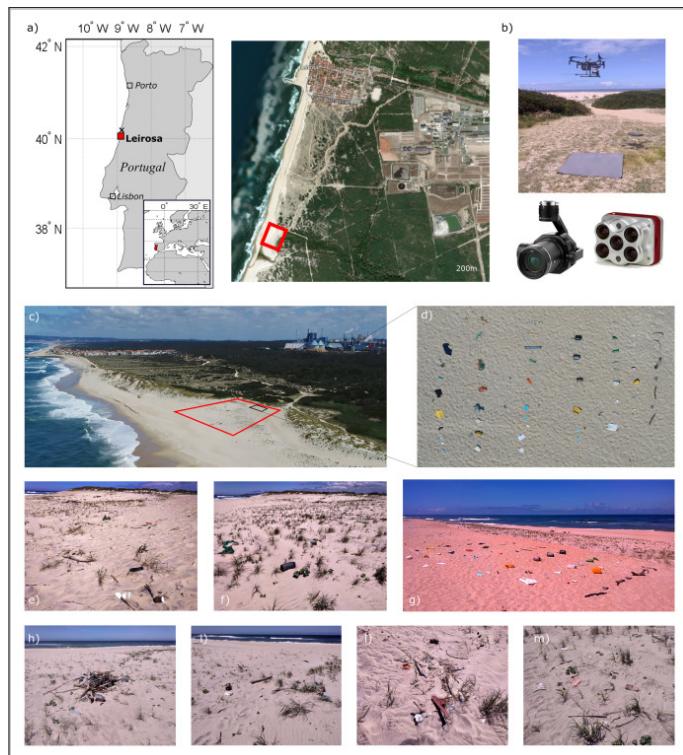
[6]-ben Sentinel-2 műholdfelvételeken tanítottak be egy Random Forest modellt azzal a céllal, hogy egy indonéziai folyóban detektáljanak hulladékot. A cikkben bevezetik az "Adjusted Plastic Index"-et, mellyel a vegetáció, föld és épületek közötti zajt csökkentik. Ennek az indexnek a kiszámításához a Sentinel-2 műhold piros, közel infravörös (NIR), illetve rövid hullámhosszú közelí infravörös (SWIR) sávokat használták fel. Validációjának Pleiades műholdképeket és drónfelvételeket klasszifikáltak Mahalanobis távolság gépi tanulási módszerrel (2.1 ábra) . A módszer növényzeten és vízen rendre 88%, illetve 85% pontosságot ért el és épületeken, törmeléken és földön rendre 62%, 53%, illetve 21% pontossággal tudta a hulladékot detektálni. A cikk szerint az utóbbi három adattípuson azért visszafogottabbak az arányok, mert a spektrális értékei az épületeknek, a földnek és a törmeléknak nagyon hasonlítanak.

## 2. Kapcsolódó kutatások



2.1. ábra. Hulladékdetektálás "Adjusted Plastic Index", Random Forest és Mahalanobis távolság segítségével [6]

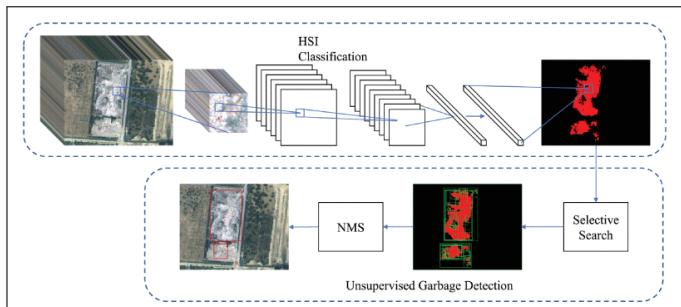
[7]-ben Spectral Angle Mapping módszert alkalmaztak multispektrális drónfelvételeken, egy Portugál tengerparton. A célja a kutatásnak az volt, hogy a tengerparton kimosott hulladékot detektálják és klasszifikálják. A módszer alkalmazásához referencia értékeket állítottak elő azzal, hogy elhelyeztek különböző anyagokból álló hulladékot a homokba, és ezekről drónfelvételt készítettek (2.2 ábra). Ezzel a módszerrel képesek voltak detektálni és klasszifikálni nem csak homok fölött található hulladékot, hanem a homokban félig elásott hulladékot is. A 472 kézzel előállított tesztadatból volt a 268 True Positive(57% összesen), 96 volt a False Positive és 204 volt a False Negative.



2.2. ábra. Spectral angle mapping referencia adatainak előállítása [7]

[8]-ben mezőgazdasági hulladékdetektálásra használtak egy Support Vector Machine modellt, Landsat 8 műholdfelvételeken. A szenzor Kék, Zöld, Piros, NIR, SWIR 1, SWIR 2 és CIRRUS sávját használták a tanítóadatok és tesztadatok előállítására. Ezután véletlenszerűen szétválasztották az adatokat tanítóadatokra és tesztadatokra. A következő osztályokra bontották az adatokat: Hálók, műanyag takarók, föld, növényzet, gyümölcsöskert, olajfás kert, város, fa, fás föld. A modell a tesztadatokat összességében 94%-os pontossággal tudta klasszifikálni, ahol a legrosszabb arányokat az olajfás kert érte el 77.78%-os pontossággal.

[9]-ben a hiperspektrális adatokon tanítottak be egy felügyelt és egy felügyeletlen gépi tanulási módszert. A működési elv az, hogy a felügyelt módszerrel klasszifikálják a hulladékkal szennyezett területeket, míg a felügyeletlen módszerrel megbecsülik a hulladékkal szennyezett terület kerületét(2.3 ábra). Ők 99.89%-os kappa együtthatóval tudtak hulladékot detektálni .



2.3. ábra. Az MSCNN működési elve [9]

## 2.0.2. Random Forest

A Random Forest egy felügyelt gépi tanulási módszer, mely magas pontossággal tud rátanulni a tanítóadatokra (túltanulás nélkül), és jól kezeli a zajt [4]. A módszer több, különbözően paraméterezett döntési fa felépítéséből kapta a nevét: miután felépítettük ezt a "döntési fa erdőt", az adatokat úgy lehet osztályozni, hogy egy többségi szavazást hajtunk végre az összes fa eredménye szerint. A fák felépítéséhez több stratégia létezik, és ezen stratégiák alapján lehet finomhangolni a modell pontosságát és méretét. Az utóbbit fontos szem előtt tartani, tekintve arra, hogy elég sok adat esetén a modell mérete lényegesen megnőhet helyes paraméterezés hiányában. Az ilyen paraméterek például a fák maximum mélysége, a fáknak átadott részadathalmaz dimenziói, egy csúcs kettévalasztásának a kritériumjai, a tanítóadatok súlyai stb. A kutatásomban ezt a modellt tanítom be, illetve paraméterezem azzal a céllal, hogy megbízható klasszifikációt tudjon

Átnézni  
mégegyszer  
a cikket és  
átvizsgálni  
az adatok  
helyességét

biztosítani. A modell minden képkockát osztályozni fog, így a bemeneti adatok az adott terület spektrális sávjai, illetve indexei.

## **3. fejezet**

### **Elemzés és tervezés**

#### **3.0.1. Kutatási cél**

A cél az, hogy a kutatás során szerzett modell megbízhatóan detektáljon hulladéklerakókat általánosan folyók mentén. Ehhez a false positive arányok minél kisebbek kell legyenek, míg a true positive arányok minél nagyobbak. Ugyanakkor nem jelent ugyanakkora problémát egy false negative, mint egy false positive, mivel a false positive eredmények fölöslegesen rossz irányba küldhetik a folyómentő csapatot. A kutatólabor 2023-as cikkjében bemutatott modell (továbbiakban meglevő modell) egyik problémája a nagy false positive arányok voltak. A modell a pusztazámor hulladéklerakóról, illetve a kiskörei víztárolóról szerzett adatokkal volt betanítva. Ezért érdemes első körben egy nagyobb adathalmazzal betanítani a modellt.

#### **3.0.2. Műhold specifikációk**

Az új Random Forest modell a PlanetScope műholdakra lesz specializálva, azon belül is a legújabb PSB.SD szenzorokra[10]. A modell számára elérhető lesz a Vörös (Red), Kék (Blue), Zöld (Green), és a közeli infravörös (NIR) sáv. A PlanetScope műholdak körülbelül 3 méter/pixel felbontással rendelkeznek [11].

#### **3.0.3. Használt indexek**

A kutatás során felhasználom a kutatólaborban már számolt indexeket. Pontosabban a Plastic Index (3.1 képlet), Normalized Difference Water Index (3.2 képlet), Normalized Difference Vegetation Index (3.3 képlet), Reversed Normalized Difference Vegetation Index (3.4 képlet), Simple Ratio (3.5 képlet) indexek kerülnek használatra [12, 5].

$$PlasticIndex(PI) = \frac{NIR}{NIR + Red} \quad (3.1)$$

$$NormalizedDifferenceWaterIndex(NDWI) = \frac{Green - NIR}{Green + NIR} \quad (3.2)$$

$$NormalizedDifferenceVegetationIndex(NDVI) = \frac{NIR - Red}{NIR + Red} \quad (3.3)$$

$$ReversedNormalizedDifferenceVegetationIndex(RNDVI) = \frac{Red - NIR}{Red + NIR} \quad (3.4)$$

$$SimpleRatio(SR) = \frac{NIR}{Red} \quad (3.5)$$

## **4. fejezet**

### **Betanítás**

#### **4.0.1. Tanítóadatok**

A betanításhoz 29 romániai hulladéklerakó és közvetlen környezete került a tanítóadatok közé, illetve a Kiskörei víztároló is. A romániai hulladéklerakókat egy helyi weboldalon lehet megtalálni, a hozzájuk tartozó koordinátákkal együtt [13]. Az ott bemutatott 46 hulladéklerakó közül 29 volt alkalmas tanításra: sok hulladéklerakó be lett tömve, vagy föld alatt működik. minden hulladéklerakóhoz letöltöttem egy-egy nyári+tavaszi, téli és őszi multispektrális műholdképet, melyeket kézzel annotáltam. A nyári és tavaszi képeket azért vontam egybe, mivel ezek hulladékdetectálás szempontjából hasonló adatokat eredményeztek. A tanítóadatok pixelenként vannak előállítva, így a végső adathalmaz 27 millió tanítóadatból (pixelből) áll. minden pixelhez hozzá van rendelve a vörös, kék, zöld, közel infravörös sáv, illetve a "PI", "NDWI", "NDVI", "RNDVI", "SR" indexek. Ezen felül minden pixel címkézve van a 4.1 táblázatban leírtak szerint. Nagyon fontos, hogy minél pontosabban meg lehessen állapítani, hogy a hulladéklerakókon mely területeken található hulladék, hiszen sok hulladéklerakón törmeléket is tárolnak, ezt általában egy külön területen. Emiatt a tanítóadatokat a következő módszerrel állítottam elő: Megvizsgáltam Google Maps segítségével [14], hogy az adott hulladéklerakónál hol tárolnak törmeléket és hol tárolnak műanyag alapú hulladékot. A Google Maps légi felvételei elég magas felbontással rendelkeznek ahhoz, hogy szemmel meglehessen különböztetni a műanyag alapú hulladékot a törmeléktől. A 4.1 ábrából látható, hogy míg a törmelék inkább fehér színt tartalmaz, addig a műanyag alapú hulladék picit vörösebb, tekintve arra, hogy a műanyag sokszor színezve van. Emellett, a hulladéklerakókat nagyon konzervatívan jelöltettem ki: csak akkor jelöltettem be egy területet, mint hulladékos terület, ha a felvételből egyértelmű volt, hogy az adott

terület műanyag alapú hulladékkel szennyezett.

Címke azonosító	Címke neve	Címke magyarázat
100	Hulladék	Azon területek, melyeken hulladékot találtunk.
200	Víz	olyan területek, melyeken kizárolag vizet találtunk, általában folyók.
300	Legelők/Erdők	Zöld övezetből álló vad területek. Ezek lehetnek fák lombjai vagy füves zónák.
400	Mezők	Olyan földes területek, melyek meg vannak művelve, illetve ahol mezőgazdasági növények találhatóak, például gabonafélék.
500	Ismeretlen	Olyan területek, melyek a korábbi kategóriákba nem sorolhatók bele. Ilyenek az épületek, aszfaltozott utak, háztetők, mezei utak.

4.1. táblázat. A tanítóadatok címkéi



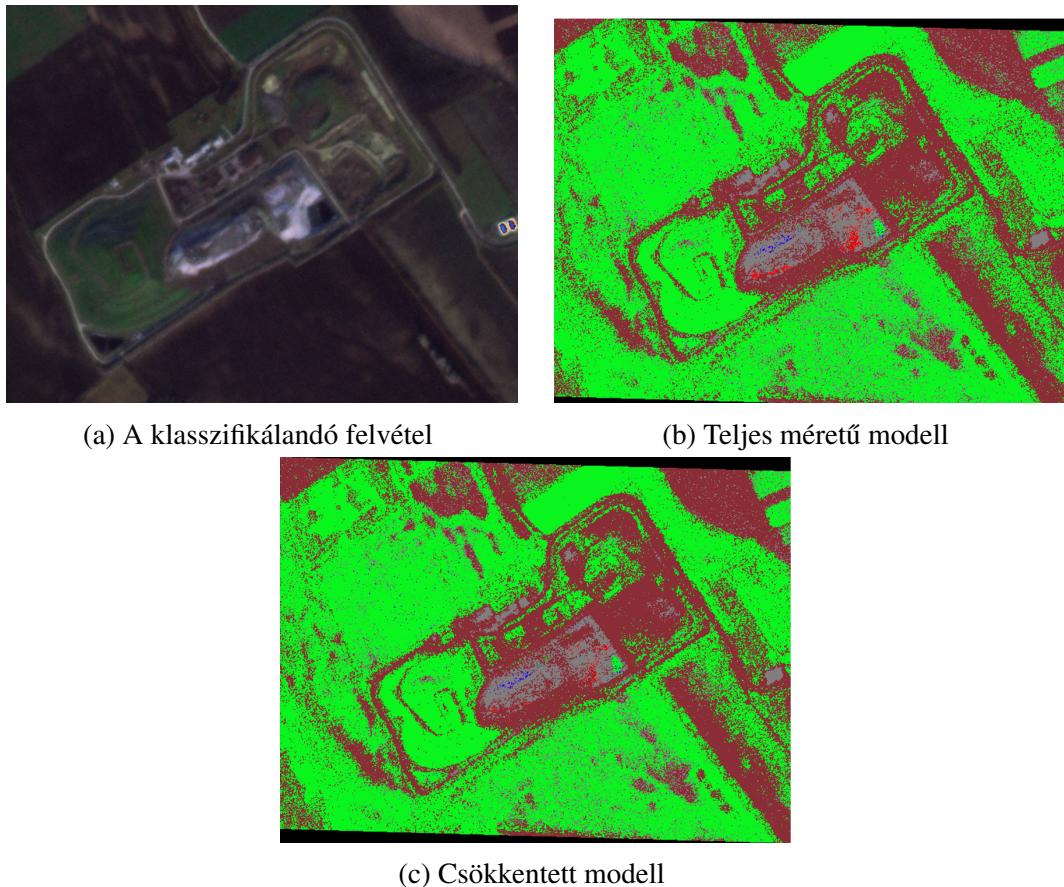
4.1. ábra. A műanyag alapú hulladék törmelék mellé helyezve. Forrás: Google Maps

#### 4.0.2. Tanítási paraméterek

A nagy adathalmaz miatt a Random Forest modell is nagyon nagy lesz (körülbelül 14GB), ami egy nehezen kezelhető méret, így érdemes módosítani a modell paraméterein, hogy ez kisebb méretű legyen. A legjobb eredményeket azzal értem el, hogy a Random Forest fák méretét 20 mélységrére limitáltam. Ennek köszönhetően a modellek méretét 2GB-ra tudtam csökkenteni, és a 4.2 ábrából látható, hogy a csökkentett modellben

táblázat a fák méretéről, a modellek méretéről és a különböző mélységekről

enyhén megnő a hulladékra vonatkozó false-negative ráta, míg a false positive arány nem nő, de cserében egy kezelhető méretű modellt kapunk.



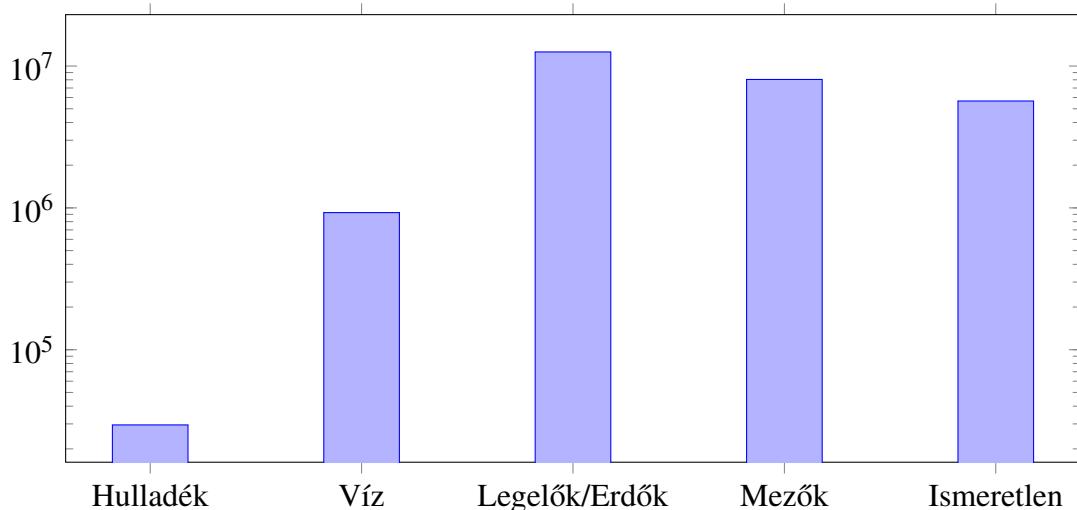
4.2. ábra. A csökkentett modell hasonlóan teljesít a teljes méretű modellhez

Továbbiakban felmerült az a probléma is, hogy a tanítóadatok nagyon aránytalanok voltak: A 4.3 ábrából látható, hogy nagyságrendekkel kevesebb adattal rendelkeztünk hulladékról, mint az összes többi adatról. Emiatt a modell nagyon sok false-negatívöt termelt. Ennek korrigálására súlyokat alkalmaztam a tanítóadatokra. A súlyok kiszámolásához az összes címkére a 4.1 képletet használtam.

$$\text{címke súlya} = \frac{\text{adathalmaz mérete}}{\text{címke darabszáma}} \quad (4.1)$$

### 4.0.3. Főkomponens analízis (PCA)

A modell méretének a csökkentésére még megpróbáltam a főkomponens analízis (PCA) alkalmazását is [15]. A módszer többek között arra is használható, hogy egy többdimenziós adathalmazból kivonja a legfontosabb információkat egy alacsonyabb dimenziószámú adathalmazba. Az ötlet az volt, hogy a bemeneti adatok



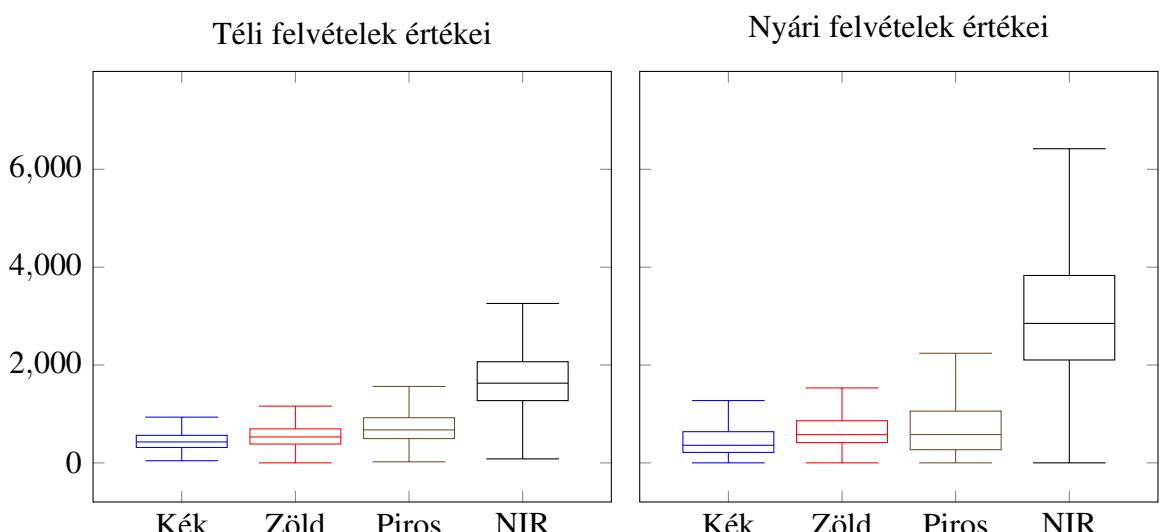
4.3. ábra. Az adatok közötti aránytalanság, logaritmikus skálázással

dimenziószámának a csökkentésével csökkenni fog a modell mérete, de érdekes módon a modell mérete nem csökkent a dimenziószám csökkentésével, helyette lényegesen megnőtt. További vizsgálatok után kiderült, hogy hogyha kevesebb dimenziójú adatot adtam a modellnek, akkor a mérete lényegesen megnőtt.

#### 4.0.4. Nyári és téli adatokra való lebontás

Alapértelmezettben a nyári és téli adatok között lényeges különbség tud lenni távérzékelés szempontból Közép-Európa területén: a téli időszakokban gyerebb a vegetáció, ködösebb a levegő, illetve a nap sem süt ugyanabból a szögből. Ez befolyásolhatja a modell pontosságát is az adott időszakokban. A nyári időszakot márciustól októberig tartó időszakként definiáltam, és a téli időszak pedig novembertől februárig tart. Az időszakok szerint vannak megválasztva, hogy mikor leveleznek ki, illetve hullatják ki a leveleiket a fák. Valóban, az októberi időszakban már inkább sárgásak lesznek a levelek, de az októberi tanítóhalmaz mérete önmagában igen kicsi érdemi tanításra. A 4.4 ábrából látható, hogy főleg a közelí infravörös (NIR) sávokon nagy eltérések vannak a nyári és téli felvételek között. Ennek fényében betanítottam külön egy

nyári és egy téli modellt, melyek teljesítményét a ?? fejezetben részletezem.



4.4. ábra. Nyári és téli adatok összehasonlítása

## **5. fejezet**

### **Tesztelés**

#### **5.0.1. A teszthalmaz**

A teszthalmaz három drinai felvételből áll, amin kézzel annotáltam a hulladékkal szennyezett területeket. Ez a terület egyben egy szárazföldi hulladéklerakót, illetve egy vízfelszíni hulladékszigetet is tartalmaz, így alkalmas minden detektálásnak a tesztelésére. A 5.1 ábrából látható, hogy Drinán úgy fogják meg az úszó műanyag-alapú hulladékot, hogy egy zsinorra ráhúznak üres hordókat, melyek a víz felszínén lebegnek. Így, minden, ami elég könnyű ahhoz, hogy a folyó felszínen ússzon (műanyagpalackok, kisebb fadarabok) megakad a hordók mögött, mígy például nagyobb fadarabok, vagy más, nehezebb uszadékok a zsinor alatt elúsznak. Így a folyó felszínén kialakuló sziget nagy koncentrációban tartalmaz műanyag alapú hulladékot, tehát alkalmas arra, hogy a modellt ezen validáljam vízfelszíni hulladékdetectáláshoz. Ráadásul erről a területről nem készültek tesztadatok ebben a kutatásban, így a modell teljesítménye az itteni felvételeken jól tesztelhető. A 5.2 ábrából látható egy-egy vizuális összehasonlítás a régi és az új modell klasszifikációja között a teszthalmaz egyik felvételén. Látszik ezen a példán, hogy az új modell több false negative-ot termel főleg a hulladéksziget körül, de ugyanakkor lényegesen lecsökkenti a false positive-ok arányát a régi modellhez képest. Ráadásul a folyó mellett található hulladéklerakót is megtalálja az új modell, míg a régi modell nem találja meg, ellenben a lerakó környékét és az utakat, épületeket gyakran hulladéknak detectálja. Ez egy fontos eredmény, hiszen amint a 3.0.1 fejezetben is tárgyaltam, célja ennek a kutatásnak, hogy csökkentsem a modell false positive arányait, miközben továbbra is meg tudja találni a hulladéklerakókat, illetve hulladékszigeteket.



5.1. ábra. A drinai hulladéksziget. Egy lebegő zsinor fogja meg a műanyagpalackokat [16]



(a) A teszthalmaz kézi annotációja

(b) A teszthalmaz annotációja a régi modellel



(c) A teszthalmaz annotációja az új modellel

5.2. ábra. Az új modell összehasonlítása a régi modellel az egyik teszt felvételen

### 5.0.2. Teljesítmény mérési módszerek

A teszthalmaz eredményekeit a "Confusion Matrix" módszerével értékeltem ki [17]. Ezután ezeket az értékeket arra használtam, hogy a "Comission rate" (5.1 képlet), "Omission rate" (5.2 képlet), "Match rate" (5.3 képlet), illetve "Extraction rate" (5.4 képlet) értékeket számítsam ki [18].

$$\text{Comission rate} = \frac{N_{com}}{N_{ref}} \quad (5.1)$$

$$\text{Omission rate} = \frac{N_{om}}{N_{ext}} \quad (5.2)$$

$$\text{Match rate} = \frac{N_{match}}{N_{ref}} \quad (5.3)$$

$$\text{Extraction rate} = \frac{N_{ext}}{N_{ref}} \quad (5.4)$$

$N_{com}$ ,  $N_{om}$ ,  $N_{match}$ ,  $N_{ext}$ ,  $N_{ref}$ , rendre a false positive, false negative (A mátrix mellékátlói), true positive (A mátrix főátlója), a modell által detektált pozitív, illetve a referencia adatokban található pozitív értékek. Összehasonlítottam az új modell teljesítményét a régi modell teljesítményével. A 5.1 táblázatból látható a két modell teljesítményének az átlaga a három felvételen.

Mérés azonosító	Régi modell átlagai (%)	új modell átlagai (%)
Comission Rate	63.67	28.13
Omission Rate	26.21	70.67
Match Rate	73.79	29.32
Extraction Rate	208.18	41.31

5.1. táblázat. A régi modell és az új modell teszteredményei átlagolva

Az új modell egy jóval kisebb false positive aránnyal rendelkezik mint a régi modell, de cserében a false-negative arányok is nagyok. Ennek oka a 5.2 ábrából látható, hiszen a régi modell sokkal több pontot detektál a hulladékszigeten, míg az új modell kevesebb pontot detektál, de továbbra is nagy mértékben megtalálja a hulladékszigetet. Illetve a 5.0.1 fejezetben is tárgyaltam, hogy a szárazföldi hulladéklerakót a folyó mellett az új modell már megtalálja, míg a régi nem találja meg. Tekintve arra, hogy a match rate a true positive-al arányos, és az Extraction rate az összes pozitive-al arányos, ezek az értékek is kisebbek lesznek, mint a régi modell értékei.

## 6. fejezet

# Megvalósítás és alkalmazás

### 6.0.1. A meglevő alkalmazások bővítése

#### Az asztali alkalmazás bővítése

A meglevő asztali alkalmazás alkalmas volt a tanítóadatok hatékony előállítására, de utólag nem lehetett visszanézni, hogy adott műholdfelvételhez milyen tanítóadatok tartoznak, illetve azt sem, hogy az adott tanítóadat hol volt mintavételezve. Az alkalmazás eredetileg egy CSV fájlból [19] tárolta el az összes pixel spektrális értékeit és indexeit, és ezt lehetett használni tanításra. Ennek az volt a hátránya, hogy nehéz volt áttekinteni illetve kiegészíteni az adatokat. Ezért az asztali alkalmazást kiegészítettem ezzel a funkcionálitással, a tanítóadatok előállítása elmentésekor az alkalmazás létrehoz egy külön raszteres réteget is külön minden műholdfelvételhez, melyen látható, hogy mely területek voltak hozzáadva a tanítóadatok közé, így tetszőleges módon előállítható/ellenőrizhető a tanítóhalmaz.

#### A szerveralkalmazás bővítése

A szerveralkalmazás és webalkalmazás is bővítésre került: a szerveralkalmazás mostmár több modellt is le tud futtatni a letöltött műholdfelvételeken és ezeket külön tárolja. A webalkalmazás mostmár képes letölteni külön ezeket az eredményeket és több hulladékmászkoló módszer eredményét is meg tudja jeleníteni, ennek köszönhetően ezeket egymással össze lehet könnyen hasonlítani valós tesztadatokon.

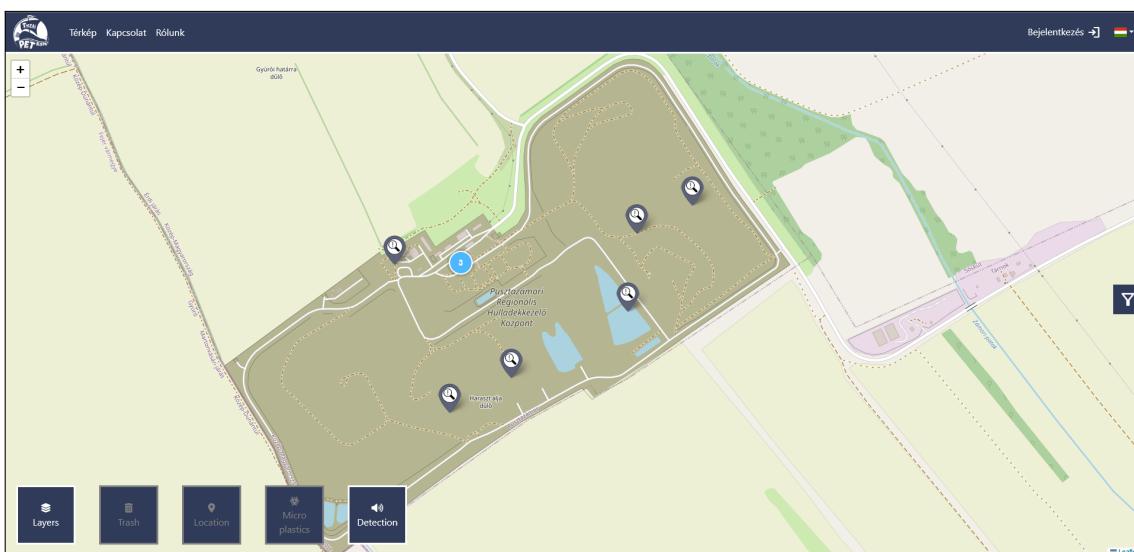
### 6.0.2. A Tiszta-Tisza alkalmazás

melyik link  
kerüljön ide?

A Tiszta-Tisza webalkalmazás a PET Kupa által használt webalkalmazás, melynek az

a célja, hogy egy olyan felületet biztosítson, ahol meglehet tekinteni a jelenleg ismert folyómentén található hulladéklerakókat, illetve akár a regisztrált felhasználók is be tudnak jelenteni ilyet. A PET Kupa megbízta az egyetemet azzal a feladattal, hogy ezt továbfejlessze, és a feladatok közé tartozott az is, hogy a Random Forest modell eredményeit integráljuk ebbe az alkalmazásba. Ezt a feladatot én vállaltam el.

Tekintve arra, hogy a Tisza-Tisza térképén pontok vannak megjelenítve, a modell által detektált területeket is pontokkal jelöljük. Ehhez egy nagyobb terület közepére helyezünk el egy pontot. Előfordulhat olyan is, hogy a modell olyan képeket klasszifikál, melyek el vannak torzítva (például magas páratartalom miatt). Ilyenkor a false-positive-ok aránya lényegesen megnő. Ennek korrigálására a Tisza-Tisza alkalmazásban a legutolsó három detektálást (legfeljebb 1 hónap különbséggel) veszem figyelembe és két kép közös metszetével döntöm el, hogy milyen területek kerülnek fel a térképre. A 6.1 ábrán láthatóak a hulladékdetektálás során előállított pontok a térképen, melyeket egy nagyítóval vannak jelölve, melyben van egy felkiáltójel. A Hulladékdetektálás konverzióját a térképen látható pontokká a 6.0.3 fejezetben részletezem.



6.1. ábra. A hulladékdetektálás által megjelenített pontok a térképen.

### A képfeldolgozás gyorsítása

A meglevő képfeldolgozó algoritmuson gyorsítani kellett, hogy elfogadható időn belül tudja feldolgozni a tanítóadatokat, illetve a naponta letöltött műholdfelvételeket a szerveren. Ezt a kutatólabor korábbi cikkjében párhuzamosítással javasolták, de egy egyszálú megoldással lényegesen tudtam gyorsítani a képfeldolgozáson: Ehhez egy nagyon hatékony Python programcsomagot, a Numpy-t [20] használtam fel, mellyel

lényegesen megnövelte a feldolgozás sebességét: A tanítóadatok feldolgozásakor a régi módszer (6.1 forráskód) 16 felvételt tudott feldolgozni 4 nap és 10 óra alatt, míg az átírt módszer (6.2 forráskód) feldolgozott 85 felvételt 20 perc alatt.

```

1 def _calculate_index(numerator: np.ndarray, denominator: np.ndarray
2 ) -> np.ndarray:
3     """
4         Calculating an index based on given numerator and denominator.
5
6         :param numerator: numerator matrix
7         :param denominator: denominator matrix
8         :return: result matrix, containing the calculated values
9         """
10
11    # variables
12    rows = numerator.shape[0]
13    cols = numerator.shape[1]
14    index = np.ndarray(
15        shape=numerator.shape,
16        dtype="float32",
17    )
18
19    # calculate index
20    for i in range(rows):
21        for j in range(cols):
22            if np.isnan(numerator[i, j]) or np.isnan(denominator[i,
23                j]):
24                index[i, j] = float("NaN")
25            elif denominator[i, j] != 0:
26                index[i, j] = numerator[i, j] / denominator[i, j]
27            else:
28                if numerator[i, j] < 0:
29                    index[i, j] = np.nanmin(numerator)
30                elif numerator[i, j] > 0:
31                    index[i, j] = np.nanmax(numerator)
32                else:
33                    index[i, j] = float("NaN")
34
35    # return index values
36    return index

```

---

### 6.1. forráskód. A képfeldolgozás régi módszere

---

```
1 def calculate_index(numerator: np.ndarray, denominator: np.ndarray)
2     -> np.ndarray:
3         """
4             Calculating an index based on given numerator and denominator.
5
6             :param numerator: numerator matrix
7             :param denominator: denominator matrix
8             :return: result matrix, containing the calculated values
9             """
10
11     # variables
12     index = np.ndarray(
13         shape=numerator.shape,
14         dtype="float32",
15     )
16
17     numerator_nan_min = np.nanmin(numerator)
18     numerator_nan_max = np.nanmax(numerator)
19
20     # calculate index
21     nan_mask = np.isnan(numerator) | np.isnan(denominator)
22     numerator_zero_mask = numerator == 0
23     denominator_zero_mask = denominator == 0
24
25     invalid_mask = nan_mask | (numerator_zero_mask &
26                                denominator_zero_mask)
27     valid_mask = np.logical_not(invalid_mask)
28
29     valid_denominator_non_zero_mask = valid_mask & np.logical_not(
30         denominator_zero_mask)
31     valid_denominator_zero_mask = valid_mask &
32         denominator_zero_mask
33
34     numerator_positive_denominator_zero_mask =
35         valid_denominator_zero_mask & (numerator > 0)
36     numerator_negative_denominator_zero_mask =
37         valid_denominator_zero_mask & (numerator < 0)
```

```

33     index[invalid_mask] = float("NaN")
34     index[numerator_positive_denominator_zero_mask] =
35         numerator_nan_max
36     index[numerator_negative_denominator_zero_mask] =
37         numerator_nan_min
38     index[valid_denominator_non_zero_mask] = (
39         numerator[valid_denominator_non_zero_mask] / denominator[
40             valid_denominator_non_zero_mask]
41     )
42
43     # return index values
44     return index

```

6.2. forráskód. A képfeldolgozás numpy-al

### 6.0.3. Közös metszet

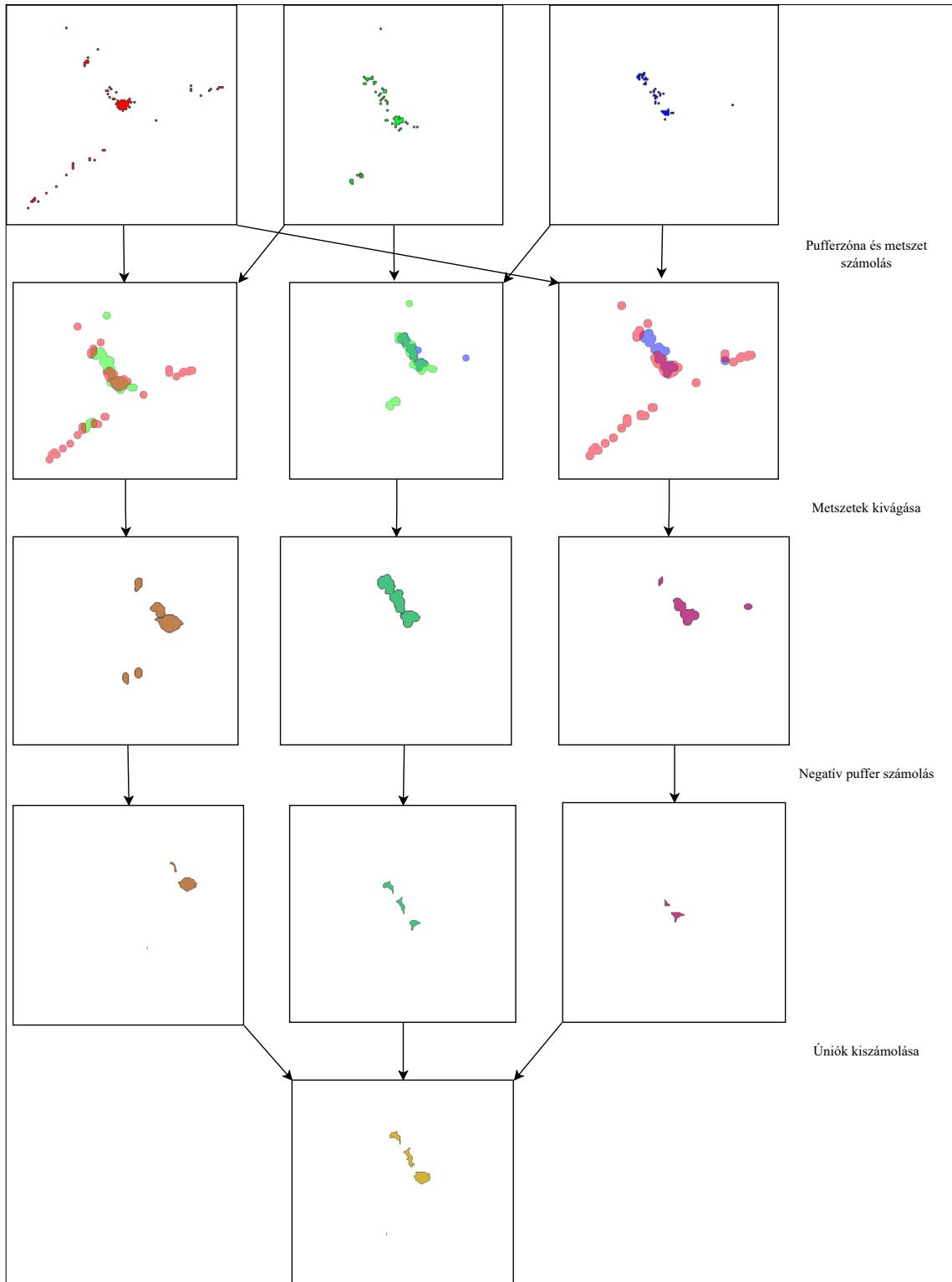
A már meglevő szervertől poligonok formájában, GeoJSON-ben [21] lehet lekérni az adott napon detektált hulladékos területeket. így érdemes poligonok metszetében kigondolni a többségi szavazást. Jelöljük BUF(P,n)-vel egy multipolygon pufferét, ahol

$$P \in \mathbb{P}$$

egy multipolygon, és n egy egész szám. Ekkor a többségi szavazást három képre a 6.1 képlet szerint lehet alkalmazni. Ezután a poligonok egy-egy belső pontját megválasztva megtudjuk jelölni a hulladéklerakókat. A megközelítés szavakra bontva a következő: az összes multipolygon párra vesszük a két multipolygon metszetét, (ahol egy n méretű hibahatár is belefér a metszet számolásba), utána a poligonpárokat rendre összeuniózzuk. A 6.2 ábra vizualizálja a képlet lépésein.

$$\text{Eredmény multipolygon} = \bigcup_{P_1 \in \mathbb{P}} \bigcup_{P_2 \in \mathbb{P}} \text{BUF}(\text{BUF}(P_1, n) \cap \text{BUF}(P_2, n), -n) \quad (6.1)$$

hasonló  
képleteket  
mások is  
ismernek  
fórumokban,  
de hivatalos  
forrással nem  
találkoztam



6.2. ábra. A 6.1 képlet lépései

## **7. fejezet**

### **Összefoglalás és eredmények**

# **Köszönetnyilvánítás**

Amennyiben a TDK projekted pénzügyi támogatást kapott egy projektből vagy az egyetemtől, jellemzően kötelező feltüntetni a dolgozatban is. A dolgozat elkészítéséhez segítséget nyújtó oktatók, hallgatótársak, kollégák felé is nyilvánítható külön köszönet.

## **A. függelék**

### **Szimulációs eredmények**

# Irodalomjegyzék

- [1] M.G. Kibria, N.I. Masuk és R. et al. Safayet. „Plastic Waste: Challenges and Opportunities to Mitigate Pollution and Effective Management”. *International Journal of Environmental Research* 17.20 (2023. jan.). ISSN: 2008-2034. URL: <https://doi.org/10.1007/s41742-023-00507-z>.
- [2] Björn Nyberg és tsai. „Leaving a plastic legacy: Current and future scenarios for mismanaged plastic waste in rivers”. *Science of The Total Environment* 869 (2023), 161821. old. ISSN: 0048-9697. DOI: <https://doi.org/10.1016/j.scitotenv.2023.161821>. URL: <https://www.sciencedirect.com/science/article/pii/S0048969723004369>.
- [3] Tim H. M. van Emmerik és tsai. „River plastic transport and deposition amplified by extreme flood”. *Nature Water* 1.6 (2023), 514–522. old. ISSN: 2731-6084. DOI: [10.1038/s44221-023-00092-7](https://doi.org/10.1038/s44221-023-00092-7). URL: <https://doi.org/10.1038/s44221-023-00092-7>.
- [4] Leo Breiman. „Random Forests”. *Machine Learning* 45.1 (2001), 5–32. old. ISSN: 1573-0565. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324). URL: <https://doi.org/10.1023/A:1010933404324>.
- [5] Dávid Magyar és tsai. „Waste Detection and Change Analysis based on Multispectral Satellite Imagery”. 2023. jan. DOI: [10.48550/arXiv.2303.14521](https://arxiv.org/abs/2303.14521).
- [6] Anjar Dimara Sakti és tsai. „Identification of illegally dumped plastic waste in a highly polluted river in Indonesia using Sentinel-2 satellite imagery”. *Scientific Reports* 13.1 (2023), 5039. old. ISSN: 2045-2322. DOI: [10.1038/s41598-023-32087-5](https://doi.org/10.1038/s41598-023-32087-5). URL: <https://doi.org/10.1038/s41598-023-32087-5>.
- [7] Gil Gonçalves és Umberto Andriolo. „Operational use of multispectral images for macro-litter mapping and categorization by Unmanned Aerial Vehicle”. *Marine Pollution Bulletin* 176 (2022), 113431. old. ISSN: 0025-326X. DOI: <https://doi.org/10.1016/j.marpolbul.2022.113431>.

- doi.org/10.1016/j.marpolbul.2022.113431. URL: <https://www.sciencedirect.com/science/article/pii/S0025326X22001138>.
- [8] Antonio Lanorte és tsai. „Agricultural plastic waste spatial estimation by Landsat 8 satellite images”. *Computers and Electronics in Agriculture* 141 (2017), 35–45. old. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2017.07.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0168169917302259>.
  - [9] Dan Zeng és tsai. „Multi-Scale CNN Based Garbage Detection of Airborne Hyperspectral Data”. *IEEE Access* 7 (2019), 104514–104527. old. DOI: 10.1109/ACCESS.2019.2932117.
  - [10] Planet.com. URL: <https://developers.planet.com/docs/apis/data/sensors/> (elérés dátuma 2024. 04. 03.).
  - [11] Planet.com. URL: <https://developers.planet.com/docs/apis/data/sensors/> (elérés dátuma 2024. 04. 03.).
  - [12] Kyriacos Themistocleous és tsai. „Investigating Detection of Floating Plastic Litter from Space Using Sentinel-2 Imagery”. *Remote Sensing* 12.16 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12162648. URL: <https://www.mdpi.com/2072-4292/12/16/2648>.
  - [13] InfoCons.ro. URL: <https://fiiunexemplu.ro/in-romania-exista-46-depozite-de-deseuri-gropi-de-gunoi/> (elérés dátuma 2024. 04. 02.).
  - [14] Google. URL: <https://www.google.com/maps/> (elérés dátuma 2024. 05. 01.).
  - [15] Hervé Abdi és Lynne J. Williams. „Principal component analysis”. *WIREs Computational Statistics* 2.4 (2010), 433–459. old. DOI: <https://doi.org/10.1002/wics.101>. eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.101>. URL: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.101>.
  - [16] EuroNews. URL: <https://www.euronews.com/green/2023/06/04/we-will-die-before-the-drina-is-clean-bosnia-villagers-hope-for-a-solution-to-polluted-riv> (elérés dátuma 2024. 05. 05.).

- [17] Russell G. Congalton. „A review of assessing the accuracy of classifications of remotely sensed data”. *Remote Sensing of Environment* 37.1 (1991), 35–46. old. ISSN: 0034-4257. DOI: [https://doi.org/10.1016/0034-4257\(91\)90048-B](https://doi.org/10.1016/0034-4257(91)90048-B). URL: <https://www.sciencedirect.com/science/article/pii/003442579190048B>.
- [18] Anett Fekete és Mate Cserep. „Tree segmentation and change detection of large urban areas based on airborne LiDAR”. *Computers & Geosciences* 156 (2021), 104900. old. ISSN: 0098-3004. DOI: <https://doi.org/10.1016/j.cageo.2021.104900>. URL: <https://www.sciencedirect.com/science/article/pii/S0098300421001928>.
- [19] Yakov Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. RFC 4180. 2005. okt. DOI: 10.17487/RFC4180. URL: <https://www.rfc-editor.org/info/rfc4180>.
- [20] Charles R. Harris és tsai. „Array programming with NumPy”. *Nature* 585.7825 (2020. szept.), 357–362. old. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [21] H. Butler és tsai. *The GeoJSON Format*. RFC 7946. 2016. aug. DOI: 10.17487/RFC7946. URL: <https://www.rfc-editor.org/info/rfc7946>.

# Ábrák jegyzéke

2.1.	Hulladékdetektálás "Adjusted Plastic Index", Random Forest és Mahalanobis távolság segítségével [6] . . . . .	6
2.2.	Spectral angle mapping referencia adatainak előállítása [7] . . . . .	6
2.3.	Az MSCNN működési elve [9] . . . . .	7
4.1.	A műanyag alapú hulladék törmelék mellé helyezve. Forrás: Google Maps	12
4.2.	A csökkentett modell hasonlóan teljesít a teljes méretű modellhez . . . . .	13
4.3.	Az adatok közötti aránytalanság, logaritmikus skálázással . . . . .	14
4.4.	Nyári és téli adatok összehasonlítása . . . . .	15
5.1.	A drinai hulladéksziget. Egy lebegő zsinor fogja meg a műanyagpalackokat [16] . . . . .	17
5.2.	Az új modell összehasonlítása a régi modellel az egyik teszt felvételel . .	17
6.1.	A hulladékdetektálás által megjelenített pontok a térképen. . . . .	20
6.2.	A 6.1 képlet lépései . . . . .	24

# Táblázatok jegyzéke

4.1. A tanítóadatok címkéi . . . . .	12
5.1. A régi modell és az új modell teszteredményei átlagolva . . . . .	18

# **Algoritmusjegyzék**

# Forráskódjegyzék

6.1. A képfeldolgozás régi módszere . . . . .	21
6.2. A képfeldolgozás numpy-al . . . . .	22